

REMARKS

Favorable reconsideration of this application, as presently amended and in light of the following discussion, is respectfully requested.

Claims 1 and 4-21 are currently pending. Claims 2 and 3 have been canceled without prejudice; and Claims 1, 4, and 19-21 have been amended by the present amendment. The changes to the claims are supported by the originally filed specification and do not add new matter.

In the outstanding Office Action, Claim 19 was rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter; Claims 1-6, 8, 9, 15, 19, and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,732,239 to Weedon et al. (hereinafter “the ‘239 patent”) in view of U.S. Patent No. 5,903,881 to Schrader et al. (hereinafter “the ‘881 patent”); Claim 21 was rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,504,899 to Raz et al. (hereinafter “the ‘899 patent”) in view of the ‘881 patent; Claims 10-13, 16, and 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘239 and ‘881 patents, further in view of the ‘899 patent; and Claims 7, 14, and 17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the ‘239 and ‘881 patents, further in view of U.S. Patent No. 6,725,341 to Peir et al. (hereinafter “the ‘341 patent”).

Applicants respectfully submit that the rejection of Claim 19 under 35 U.S.C. § 101 is rendered moot by the present amendment to Claim 19. Claim 19 has been amended to be directed to a transaction processing computer having a processor for processing a plurality of transactions, and is thus not directed to software *per se*. Accordingly, Applicants respectfully submit that the rejection of Claim 19 is rendered moot.

Amended Claim 1 is directed to a concurrency control method in a transaction processing system for processing a plurality of transactions in parallel with respect to

hierarchical data, the concurrency control method comprising: (1) producing a copy of the hierarchical data at a time of starting an access to the hierarchical data by each transaction; (2) judging whether a collision between one of reading access or writing access to be made by a first transaction with respect to a copy of the hierarchical data for the first transaction and another one of the reading access or writing access made by the second transaction with respect to a copy of the hierarchical data for the second transaction will occur or not, when the first transaction and the second transaction are accepted at the same time as concurrent transactions for accessing the same location of the hierarchical data, wherein the first transaction is started earlier than the second transaction; (3) carrying out a processing for avoiding the collision due to the concurrent transactions when the judging step judges that the collision will occur; and (4) reflecting a writing access made by the first transaction with respect to a copy of the hierarchical data for the first transaction, on the hierarchical data, when the first transaction is to be finished normally, and reflecting the writing access made by the first transaction also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet, wherein a third copy of the hierarchical data is provided in which all the transactions having been finished are reflected, and wherein when the first transaction is to make the reading access with respect to a copy of the hierarchical data, the judging step judges whether the collision will occur or not according to whether first data looked up by making the reading access with respect to the copy of the hierarchical data for the first transaction and second data looked up by making the reading access with respect to the third copy are identical or not. Claim 1 has been amended to incorporate limitations recited in Claim 3 and to further clarify that the first transaction is started earlier than the second transaction. No new matter has been added.

Regarding the rejection of Claim 1 under 35 U.S.C. § 103(a), the Office Action asserts that the ‘239 patent discloses everything in Claim 1 with the exception of the “hierarchical data limitation,” and relies on the ‘881 patent to remedy that deficiency.

The ‘239 patent is directed to a method for preventing concurrent access to an object in a database including the steps of responding to a first transaction by retrieving a cache version of the object from an exclusive access cache; responding to any concurrent transactions by retrieving respective new instances of the object from the database; completing any current transactions by selectively committing any updates that were made to the object with a write operation to the database; and completing the first transaction by selectively committing any updates that were made to the object with a write to the data store and a deposit to the exclusive access cache. In particular, the ‘239 patent discloses that “only the first transaction can ever obtain data from the exclusive access cache, but the existence of current transactions when that first one completes precludes the first from updating the exclusive access cache. Rather, the last transaction to commit can update the cache, or leave it empty as the case may be.”<sup>1</sup> Thus, the ‘239 patent discloses that writing access made by a first transaction with respect to a copy of hierarchical data for the first transaction is reflected on the hierarchical data, but not reflected on a copy of the hierarchical data for a second transaction if the second transaction is not finished yet.

Thus, Applicants respectfully submit that the ‘239 patent fails to disclose the step of reflecting writing access made by the first transaction also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet, as recited in amended Claim 1. Rather, the ‘239 patent discloses that each transaction is completed, meaning that it is either committed or rolled back, and if the transaction is committed, a copy of the record is written back to the database if it was modified. However, if the transaction was rolled back, a

---

<sup>1</sup> ‘239 patent, Abstract. Emphasis added.

copy of the record is thrown away. Thus, it is impossible to reflect the writing access on the copy that has been thrown away.

Further, Applicants respectfully submit that the '239 patent fails to disclose that a third copy of the hierarchical data is provided in which all the transactions having been finished are reflected, and that when the first transaction is to make the reading access with respect to a copy of the hierarchical data, the judging step judges whether the collision will occur or not according to whether first data looked up by making the reading access with respect to the copy of the hierarchical data for the first transaction and second data looked up by making the reading access with respect to the third copy are identical or not, as recited in amended Claim 1.<sup>2</sup>

The '881 patent is directed to a system having three simultaneously displayed items of information, including a list of transaction instructions, a list of uncleared transactions, and a list of cleared transactions. Further, the '881 patent discloses a database module that stores users' data in a combined relational-hierarchical data model that is used to organize accounts by a financial institution such that all transactions for each account are stored within the account. However, Applicants respectfully submit that the '881 patent fails to remedy the deficiencies of the '239 patent, as discussed above. In particular, the '881 patent is silent regarding reflecting the writing access made by a first transaction also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet, as recited in amended Claim 1. Further, the '881 patent fails to disclose the limitation added to Claim 1 from Claim 3.

Thus, no matter how the teachings of the '239 and '881 patents are combined, the combination does not teach or suggest the step of reflecting the writing access made by the first transaction also on a copy of the hierarchical data for the second transaction if the second

---

<sup>2</sup> Applicants refer the Examiner to the non-limiting example shown in Fig. 7, in particular, the document D-all 121.

transaction is not finished yet, as recited in amended Claim 1. Moreover, Applicants respectfully submit that the combined teachings of the '239 and '881 patents fails to disclose the third copy of the hierarchical data, and judging whether the collision will occur or not according to whether first data looked up by making the reading access with respect to the copy of the hierarchical data for the first transaction and the second data looked up by making the reading access with respect to the third copy are identical or not, as recited in Claim 1. Accordingly, Applicants respectfully submit that the rejection of Claim 1 is rendered moot by the present amendment to that claim.

Regarding the rejection of dependent Claims 7, 10-14, and 16-18 under 35 U.S.C. § 103(a), Applicants respectfully submit that the '899 and '341 patents fail to remedy the deficiencies of the '239 and '881 patents, as discussed above. In particular, regarding Claims 7, 14, and 17, Applicants note that the Office Action asserts that the '341 patent teaches an upper limit. However, Applicants note that the '341 patent merely discloses a cache management system to enhance cache efficiency in shared memory distributed cache multiprocessor computer systems. As noted by the outstanding Office Action, the '341 patent discloses that "such rules that restrict memory write backs to lines likely to be needed in other caches serve to further limit the number of transfers over the system memory bus to alleviate risk of flooding the bus with L1 cache write-back activity." Applicants respectfully submit that this passage in the '341 patent relates only to limiting the traffic on the bus while maintaining cache coherence, by deferring write back operations such that write backs from the L1 to L2 cache are not written back to memory over the system bus if the write-through bit of the written back line is not on, and that the L1 to L2 write back is not written to system memory when the written back line is not present in another cache's IHT. However, Applicants respectfully submit that this disclosure is unrelated to the upper limit to a number of shared copies that can be recorded, as recited in Claim 7. For the reasons stated above,

Applicants respectfully submit that dependent Claims 7, 10-14, and 16-18 patentably define over any proper combination of the '239, '881, '899, and '341 patents.

Claim 19 is directed to a transaction processing computer that includes a reflecting unit configured to reflect a writing access made by the first transaction also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet. Further, Claim 20 recites a fourth computer program code causing the computer to reflect the a writing access made by the first transaction also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet. As discussed above, these limitations are not suggested by the combined teachings of the '239 and '881 patents. Accordingly, for the reasons stated above, Applicants respectfully submit that the rejection of Claims 19 and 20 are rendered moot by the present amendment to Claims 19 and 20.

Claim 21 is directed to a computer program product which employs a storage medium for causing a computer to function as a transaction processing system for processing a plurality of transactions in parallel with respect to hierarchical data, the computer program product comprising: (1) a first computer program code loaded in a processor for causing the computer to accept transactions which are temporarily overlapping; (2) a second computer program code loaded in a processor for causing the computer to produce a copy of the hierarchical data at a time of starting execution of each transaction; (3) a third computer program code loaded in a processor for causing the computer to judge whether or not the copy of a first transaction will conflict with the copy of a second transaction after execution of the first and second transaction, wherein the first transaction is started earlier than the second transaction; (4) a fourth computer program code loaded in a processor for causing the computer, when it is judged that a conflict will occur between the first and second transactions, to halt the execution of the second transaction until the execution of the first transaction has been completed; and (5) a fifth computer program code loaded in a processor

for causing the computer to reflect the copy of the first transaction in the hierarchical data, when the first transaction is finished normally, and reflect the copy of the first transaction in the copy of the second transaction, when the second transaction is not finished yet, wherein the second computer program code is executed to further produce a third copy of the hierarchical data in which all the transactions having been finished are reflected, and wherein when the first transaction is to make the reading access with respect to a copy of the hierarchical data, the third computer program code is executed to judge whether the collision will occur or not according to whether first data looked up by making the reading access with respect to the copy of the hierarchical data for the first transaction and second data looked up by making the reading access with respect to the third copy are identical or not. The changes to Claim 21 are supported by the originally filed specification and do not add new matter.

The '899 patent is directed to a computer-implemented method of processing global transactions that are distributed across a computing system and local transactions that are not distributed across the computing system. In particular, the '899 patent discloses that when a second global transaction performs a read operation before a conflicting write operation of a first global transaction is committed at a time when the second global transaction has not yet committed, the second global transaction is aborted to insure that the order in which the global transactions are committed is not different from the conflict order of the global transactions. Thus, even if a data copy were provided for the second global transaction and maintained even after aborting the second global transaction, the results of the first global transaction are not reflected on the data copy provided for the second global transaction. Although the '899 system might make use of its own copy for the second transaction, the result of the first transaction is not reflected on this copy.

Thus, Applicants respectfully submit that the '899 patent fails to disclose a fifth computer program code loaded in a processor for causing the computer to reflect a copy of

the first transaction and a copy of the second transaction, when the second transaction is not finished yet, is recited in Claim 21. Rather, the '899 patent discloses that the second global transaction is aborted and does not disclose that a data copy for the second global transaction is maintained even after aborting the second global transaction, or that the result of the first transaction would be reflected in this copy.

Applicants respectfully submit, as discussed above, that the '881 patent fails to remedy the deficiencies of the '899 patent, as discussed above.

Thus, no matter how the teachings of the '899 and '881 patents are combined, the combination does not teach or suggest a fifth computer code loaded in a processor for causing the computer to reflect a copy of the first transaction in a copy of the second transaction, when the second transaction is not finished yet, as recited in Claim 21. Moreover, Applicants respectfully submit that the combined teachings of the '899 and '881 patents fails to disclose that the second computer code is executed to further produce a third copy of the hierarchical data in which all the transactions having been finished are reflected, and wherein the first transaction is used to make the reading access with respect to a copy of the hierarchical data, the third copy program code is executed to judge whether the collision will occur or not according to whether first data looked up by making the reading access with respect to the copy of the hierarchical data for the first transaction and second data looked up by making the reading access with respect to the third copy are identical or not, as recited in amended Claim 21. Accordingly, Applicants respectfully submit that the rejection of Claim 21 is rendered moot by the present amendment to that claim.

Thus, it is respectfully submitted that independent Claims 1 and 19-21 (and all associated dependent claims) patentably define over any proper combination of the cited references.

Consequently, in view of the present amendment and in light of the above discussion, the outstanding grounds for rejection are believed to have been overcome. The application as amended herewith is believed to be in condition for formal allowance. An early and favorable action to that effect is respectfully requested.

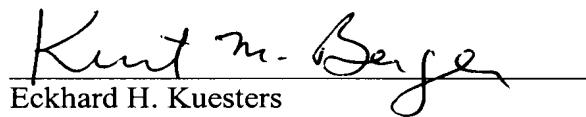
Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,  
MAIER & NEUSTADT, P.C.

Customer Number  
**22850**

Tel: (703) 413-3000  
Fax: (703) 413 -2220  
(OSMMN 08/07)  
EHK/KMB:sjh

I:\ATTY\KMB\248\S\248156US\248156US-AM 05-10-08.DOC

  
\_\_\_\_\_  
Eckhard H. Kuesters  
Attorney of Record  
Registration No. 28,870

Kurt M. Berger, Ph.D.  
Registration No. 51,461